REMARKS

A review of the application revealed incompatibilities between the specification and drawings, that the specification contained some grammatical errors. By this amendment all these errors have been corrected.

New drawings 8C and 10 and descriptions thereof have been added. The drawings and the descriptions are fully supported by the specification as originally filled. These additions were made to support added apparatus claims, which are written as means plus function for practicing the process cited in added process claims. No new matter has been added.

Claims 1 and 3-24 are in this application. Claim 2 has been canceled, claim 1 has been amended to specifically recite that the rectangles created are "boundary" rectangles, and claims 3 - 24 have been added.

Claims 7 and 8 contain all the limitations of cancelled claim 2.

All the new claims are fully supported by the specification as originally filed.

To aid the Examiner in the examination of this application, Applicant has included the amended pages with the amendment markings thereon and a new specification containing all the amendments.

Respectfully submitted,

Seymour Levine

Seymour Levine
Registration No. 27,713
9B Weavers Hill
Greenwich, CT 06831-4245

Phone (203) 532-1661
Fax (203) 532-1662

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, P. O. Box 1450 Alexandria, VA 22313-1450, on April 29, 2004 .

Seymour Levine

Seymour Levine

April 28, 2004
Date Signed

Figure 8B is a vector diagram useful for explaining the determination of counter-clock wise and clock wise turns.

Figure 8C is a block diagram of an apparatus for generating a moving haven boundary.

Figure 9 is a set of line segments useful for explaining a method for determining a start segment for generating a buffer boundary.

Figure 10 is a block diagram of an apparatus that may be utilized for the buffer generator of Figure 8C.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

In the figures elements mentioned in a figure and appear in subsequent figures bare the same reference numerals.

Refer now to Figure 1A, wherein a moving haven is shown to proceed along a first leg 11 to a waypoint 13 and then along a second leg 15, forming a boundary 17. To calculate the moving haven a first rectangle 19 is drawn as shown in Figure 1B. This rectangle has a width W equal to the width of the moving haven and a length L_1 which starts at the trailing edge 21 of the moving haven and extends to the way point 13. A second rectangle 23 is drawn as shown Figure 1C. This rectangle has the width W and a length L_2 which starts at the waypoint 13 and extends to the leading edge 25. The rectangles 19 and 23 respectively positioned along the first and second legs establish a gap 27, shown in Figure 1D, on the boundary of the moving haven. This boundary gap 27 is closed by drawing an arc 29 from the corner 19a of the rectangle 19 to the corner 23a of the rectangle 23, thus forming a pie slice having sides 29 a and 29b. The boundary 17, shown in Figure 1A, of the moving haven comprises the sides 19b and 19c of the rectangle 19, sides 23b and 23c of the rectangle 23, sections 19d and 23d of sides of the rectangles 19 and 23, which respectively extend from intersection 33 to the sides 19c and 23c, and the arc 29.

Two of the three criteria identified as requirements for success have been discussed. First, the problem is subdivided into smaller simpler sub-problems, Second, the problem avoids special case logic in that the orientation and relationship of the intermediate shapes is inconsequential since any arbitrary combination of

with a specified radius for each point on the input polygonal line requiring an arc segment. At each of these points, the arc is generated around the obtuse angle formed by the intersecting line segments. To complete the moving haven for the polygonal line shown in Figure 4A, an arc 71a must extend from the vertex 65b of rectangle 65 to the vertex 67a of the rectangle 67 and an arc 71b must extend from the vertex [67d] <u>67c</u> of the rectangle 67 to the vertex [69c] <u>69d</u> of the rectangle 69.

These arcs [are] <u>may be</u> approximated by a series of directed line segments[,]--.-- [as] As shown in Figure 5B three directed lines [are] <u>may be</u> chosen to approximate the arc, a line segment 72a from point 73a to point 73b, a line segment 72b from point 73b to point 73c, and a line segment 72c from point 73c to point 73d. Just as with the generation of rectangles, the line segments representing the arc [must] <u>should</u> be generated in a clockwise direction.

In the above illustration, line segments 72a, 72b, and 72c approximate the arc. These line segments form a clockwise rotation along the circle that defines the arc. Though only three line segments were used to illustrate the arc approximation, it should be understood that a greater number of line segments may be used to yield a smoother curve.

The following steps may be applied to approximate [of] an arc [for] <u>about a vertex of</u> an input polygonal line ABC, shown in Figure 6.

1 .A vector X_1 that connects the vertex to the beginning of the arc is determined. This will be a vector with a length equal to the specified radius, starting from vertex B, and perpendicular to AB.

2. A vector X_2 that connects the vertex B to the end of the arc is determined. This will be a vector with a length equal to the specified radius, starting from vertex B, and perpendicular to BC.

3. Add the point R_1 at the end of vector X_1 to an ordered list of points.

4. Rotate X_1 by an angle θ to establish a vector X_3. Smaller values of θ will result in a better approximation of the arc. If the total angle between the vector X_3 and the start vector X_1 is less than the angle

between the end vector and the start vector, add the point at the end of the vector X_3 to the ordered list.

5. If the total angle between X_3 and the start vector X_1 is less than the angle between the end vector and the start vector, repeat step 4 rotating the vector X_3 through the angle θ and adding the point at end of the rotated vector to the list. Otherwise proceed to step 6.

6. Add the point R_3 at the end of the end vector X_2 to the list.

If θ was set to 30 degrees for the arc shown in Figure 6, the process would generate the 4 points (73a, 73b, 73c, and 73d) shown in Figure 5B to define the arc shown in Figure 6. The final output of this step of the method would be the line segments 72a, 72b, and 72c which connect these points.

Arc representative line segments provided at the output 61a of the arc generator 61 are vectors, like the vectors representative of the rectangles, are oriented in a clockwise direction. This set of line segments and the set of line segments for the formation of the rectangles are combined in a combine sets step[.] 75. The resulting set of combined line segments sets are provided at output 75a of set combiner 75 and coupled therefrom to a start line segment determinator 77.

Before the boundary of the moving haven 78 can be established by a moving haven boundary generator 79, a starting line segment must be specified. The boundary determinator 79 receives the combined line segments from the output 75a of the combined sets step 75 and a starting line designated by the start line determinator 77. Establishment of a correct moving haven boundary requires a starting line segment which is on and has its beginning on the moving haven boundary. Referring to Figure 7, line segments 81a, 81b, 81c, having the entire segment on the boundary, and line segments 81d, 81e, having the beginnings on the boundary, are acceptable; while [81e] 81f, 81g, 81h, 81i, although having portions and their ends on the boundary, their beginnings are within the boundary, and are therefore, not acceptable. Line segment 81j is not acceptable, it is entirely within the boundary. The segments that approximate the arcs 82 and 84 are all acceptable.

3. Look for intersections between X and any other segment. In Figure 8A, the segment X intersects 83b, 83c, 83d, 83e, and 83f. Call this list of segments the candidate set of segments, named S.

4. Drop all segments from the set S whose end point touches the starting point of the starting segment X. This eliminates segment 83e.

5. Drop all segments from the set S that result in a clockwise turn, unless that segment intersects X at the end point B. This eliminates 83b, but not 83f.

6. For each of the remaining line segments in S, find where those segments intersect X. Only segments 83c, 83d, and 83f remain in S. Call the point [that] at which 83c intersects X, point Q. Notice that 83d also intersects X at Q. Call the point that 83f intersects X, point P. Of the intersection points, find the one that is closest to the start point A of X. In [Figure A] <u>Figure 8A</u>, Q is closest to A. Drop all line segments that do not include this closest intersection point. This removes 83f from S, leaving only 83c and 83d.

7. If S is empty then terminate the process.

8. Of the remaining segments in S, select the one that results in the smallest angle θ, where θ is measured as shown in Figure 8B. Call the selected segment W. In [Figure 8B] <u>Figure 8A</u>, the angle between X and 83d is smaller than the angle between X and 83c, so W would be set to 83d. 9. Let Z be the intersection of W and X. In Figure 8B Z = Q. If Z is already in L, add Z to the end of the list L and terminate the process. Otherwise, add Z to the end of the list L, and go to the next step.

IO. Set X to be the segment that starts with Z and ends with the end point of W. Repeat, starting with step 2 and using segment 86, having its beginning point at point Q and its ending point at point C, as the starting segment.

An apparatus for creating a moving haven boundary as described above is illustrated in Figure 8C. The line segments and waypoints of the voyage plan are coupled to a polygonal line generator 90, wherein the line segments of the voyage plan (vectors between waypoints) are assembled and processed as described above, with reference to Figure 8A, to provide a polygonal line representative of the voyage plan. These voyage plan line segments are coupled to a rectangle generator 92 wherein line segments (vectors) that form the rectangles about respective line segments of the polygonal line are determined. The rectangle line segments are utilized in arc generator 94 wherein arc line segments are established to approximate required arcs about a waypoint of the voyage plan polygonal line. Rectangle line segments and arc line segments are coupled to a segment combiner 96 of a boundary generator 98. Segment combiner 96 combines the rectangle line segments and arc line segments to establish a set of line segments which are coupled to start line selector 102. A start line segment is selected by the start line selector 102 which is coupled to an intersect segment selector 104 wherein a second line segment is selected that intersects the start line segment in accordance with the selection procedure previously described. The second line segment is coupled to a repeat generator 106 which designates it as a start line and causes the intersect selector 104 to select a third line from the set of line segments that intersects the second line segment in accordance with the selection criteria. The process is continued until all line segments in the set of line segments have been used. The moving haven boundary is then completed and a buffer generator 105 is activated to establish a buffer between the moving haven and the moving haven boundary.

Still referring to Figure 8C; the start line selector 102 may comprise a segment selector 108 wherein line segments in the segment combiner 96 that are entirely on or have a beginning on the moving haven boundary are selected. These selected line segments are coupled to a segment locator 110 wherein those line segments having start points at a position that is predetermined are identified and coupled to a direction determinator 112. Should more than one line segment originate at the predetermined position, the direction determinator 112 selects the one that points mostly in a predetermined direction, this direction may be "up", as previously mentioned

Once the moving haven boundary has been determined the moving haven buffer is calculated.

An overview for calculating the moving haven buffer has been previously discussed. The primary differences in applying the simple shapes for the determination of the buffer and applying the simple shapes to the determination of the moving haven boundary. are: .

A) For determining the buffer, the input polygonal line is the geometry of the moving haven boundary, rather than the voyage plan. The value used for the width is twice the buffer width, rather than the moving haven width.

B) The input polygon for arc generator 61 (Figure 3) must be altered. Specifically, a vertex must be added to the end of the input polygonal line that has the same coordinates of the second vertex of the moving haven boundary.

C) The rectangle generator only generates the portion of each buffer rectangle that is within the moving haven boundary. Also the arc generator only generates arcs within the boundary.

D) A different method must be used for determining the starting segment input to the boundary generator.

In determining the moving haven boundary, there is no need to draw an arc around the first or last vertex of the input polygon. Because the buffer is a closed polygon, it is necessary to draw an arc around each vertex. Rather than modifying the arc generator 61 (Figure 3) when creating the buffer, the input polygon is altered by adding a vertex to the end of the input polygonal line that has the same coordinates as the second vertex of the moving haven boundary. [The addition of] <u>By adding</u> the second vertex of the moving haven boundary to the end of the input polygonal line [and]- the actual end point of the boundary is no longer the last point of the input

polygonal line. As a consequence an appropriate arc will <u>be</u> drawn for each vertex of the unaltered input polygonal line.

The buffer, by definition, will never extend outside the moving haven boundary. As a consequence, any portion of any segments created by the arc or rectangle generators that are outside the boundary will never be a part of the final buffer geometry. To reduce the number of segments that the boundary generator processes, the arc generator only creates segments for arcs that are within the moving haven boundary and the rectangle generator only creates the portion of the buffer rectangle that is within the moving haven boundary, as shown in Figures 2A and 2B and described in the text relating thereto.

The method for finding the start segment to be used as input for the boundary generator 79 (Figure 3) that was used for the moving haven boundary is not appropriate for the generation of the buffer. To establish the polygonal line for the buffer having segments 91a, 91b, 91c, and 91d, illustrated in Figure 9, the following sequence of steps may be performed <u>by buffer generator 80 (Figure 3)</u>:

1) Let R be the set of segments used as inputs into the boundary generator 79 (Figure 3). R includes the segments 91a, 91b, 91c, and 91d.

2) Find a segment in R that is as least as long as all other segments in R. Segment 91a meets this criteria.

3) Find the center of 91a and label it C, so that 91a comprises two segments 93a, terminating at C, and 93b, originating at C.

4) Replace segment 91a with segments 93a and 93b.

5) Use 93b as the starting segment for the buffer bounder generation in boundary generator 79.

6) Modify the set R so it includes segments 93b, 91b, 91c, 91d, and 93a.

<u>Refer now to Figure 10, wherein a block diagram of a buffer polygonal line generator that may be utilized for the buffer generator 80 of Figure 8C is shown. The boundary polygonal line is coupled to a buffer rectangle generator 114 wherein</u>

<u>rectangles which may be used for generating the buffer polygonal line are generated. These rectangles have widths that are twice the desired buffer width, lengths that are equal to the distance between vertices of the boundary rectangles or between a vertex and the beginning of a boundary arc, and are centered on the boundary polygonal line. Rectangles generated by the rectangle generator are coupled to a buffer arc generator 119, wherein the rectangles are coupled to a first vector generator 116 and a second vector generator 118. The first vector generator establishes a vector between a vertex of an end edge of a buffer rectangle and the point at which the end edge intersects the moving haven boundary. The second vector generator establishes a vector between the end point of the leading edge of the next adjacent buffer rectangle and the point at which that leading edge intersects the moving haven boundary. These vectors are of equal length and their points of intersection coincide as shown at point 48 in Figure 2A. Vectors so established are coupled to an angle determinator 120 wherein the angle between the first and second vectors are determined. Once the angle between the first and second vectors is determined, the first vector and the angle between the first and second vectors are coupled to a vector rotator 122 wherein the first vector is rotated through selected angular increments until the entire angle between the first and second vectors has been traversed. The vectors resulting from the rotations and the first and second vectors are coupled to an end point locator 124 wherein the vector end points are determined. These end points are then coupled to a buffer polygonal line generator wherein they are connected to approximate an arc which is incorporated into the buffer polygonal line.</u>

While the invention has been described in its preferred embodiments, it is to understood that the words that have been used are words of description rather than

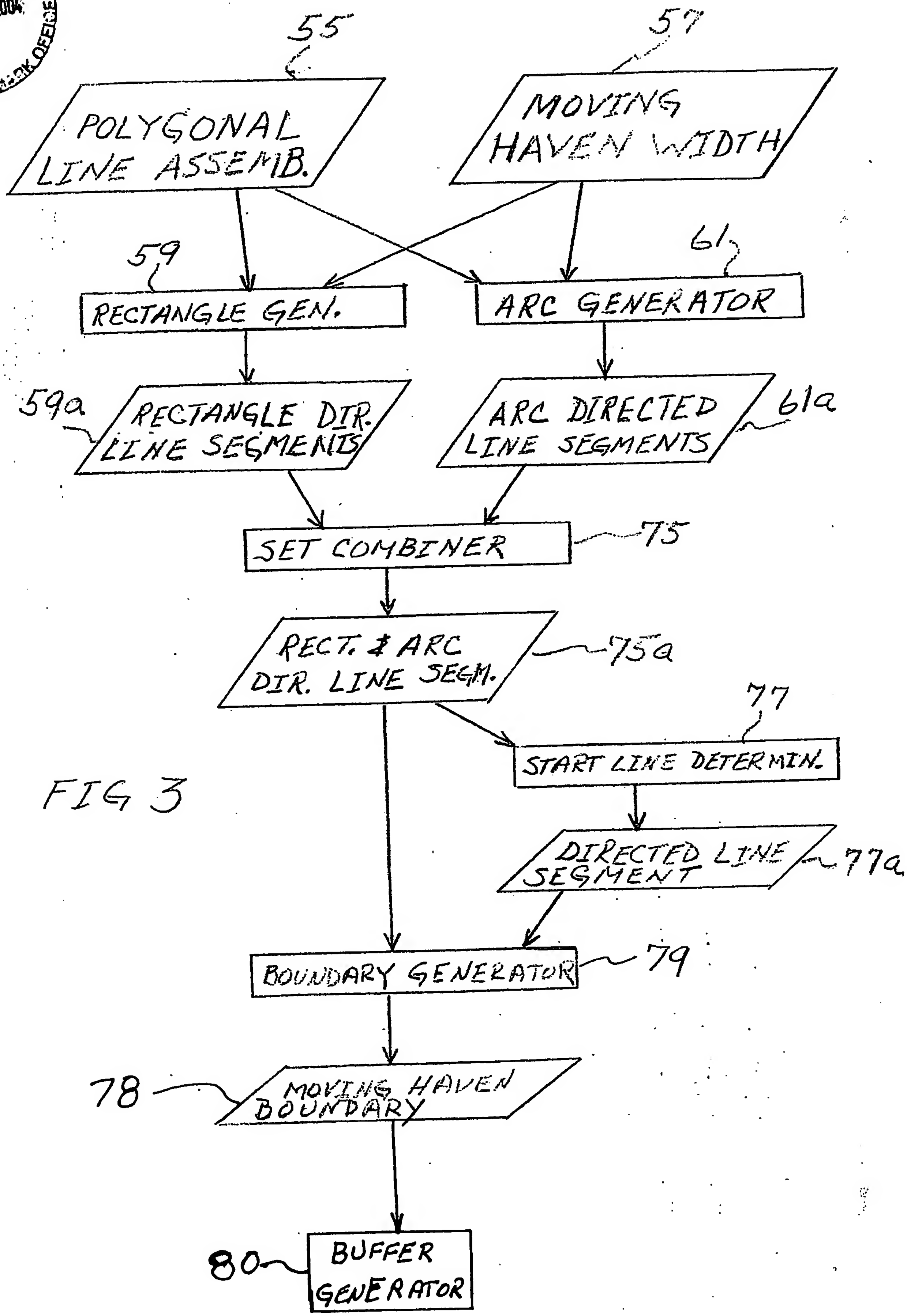
55
57
POLYGONAL LINE ASSEMB.
MOVING HAVEN WIDTH
59
61
RECTANGLE GEN.
ARC GENERATOR
59a
RECTANGLE DIR. LINE SEGMENTS
ARC DIRECTED LINE SEGMENTS
61a
SET COMBINER
75
RECT. & ARC DIR. LINE SEGM.
75a
77
START LINE DETERMIN.
DIRECTED LINE SEGMENT
77a
BOUNDARY GENERATOR
79
78
MOVING HAVEN BOUNDARY
80
BUFFER GENERATOR

FIG 3

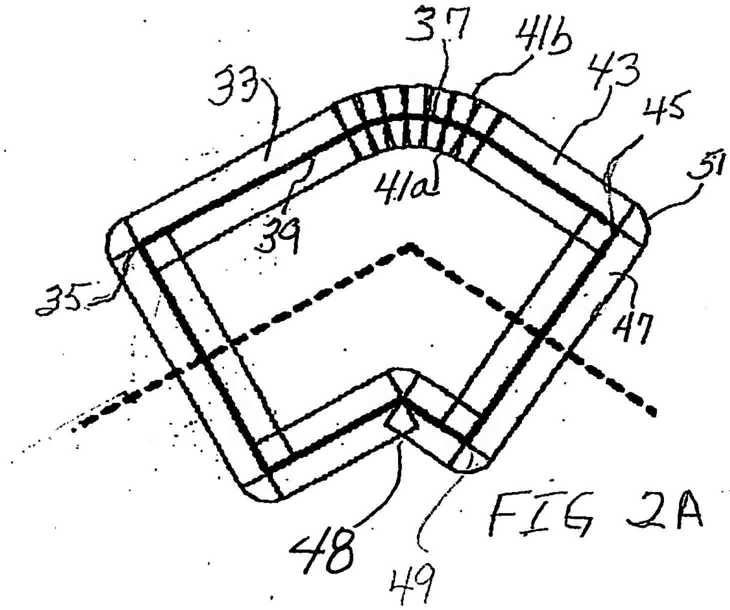

FIG 2A

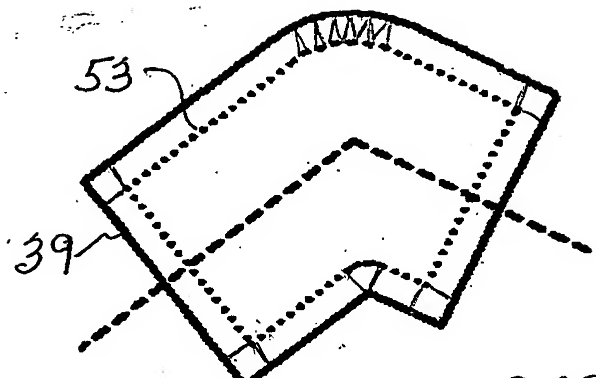

FIG 2B

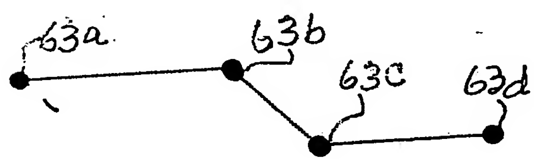

FIG 4A

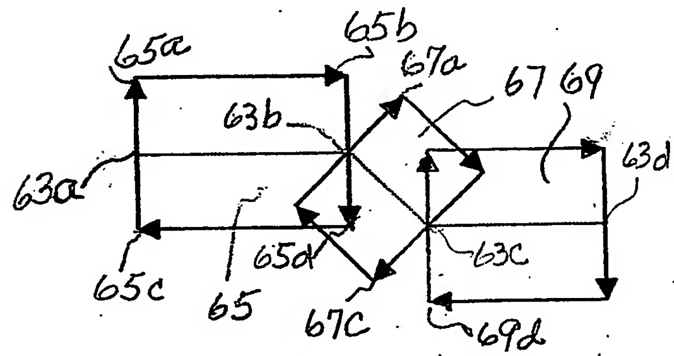

FIG 4B

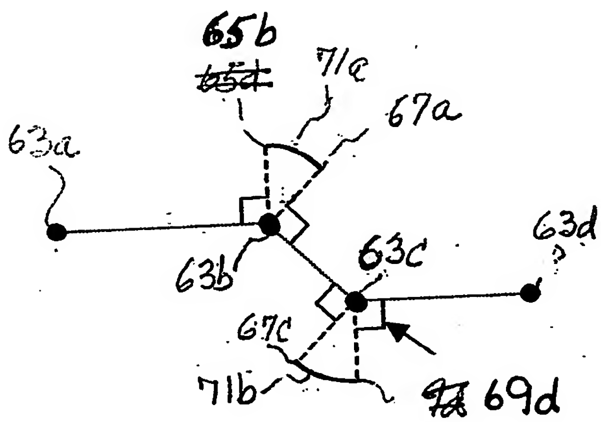

FIG 5A

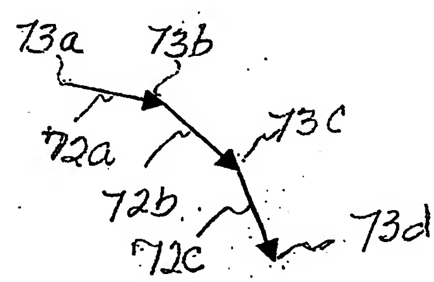

FIG 5B

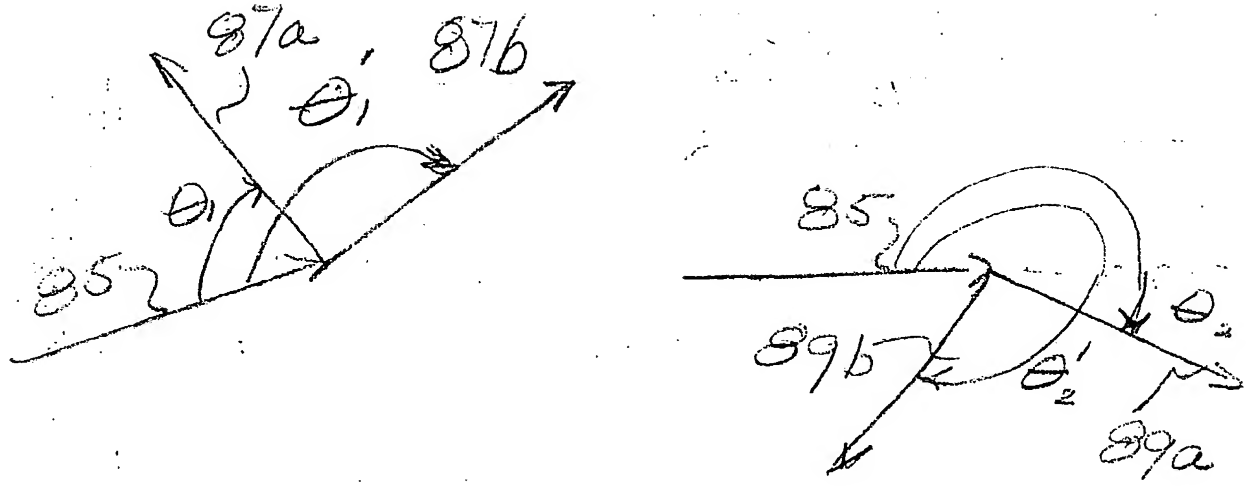
87a
87b
θ_1'
θ_1
85
85
θ_2
89b
θ_2'
89a

FIG 8B

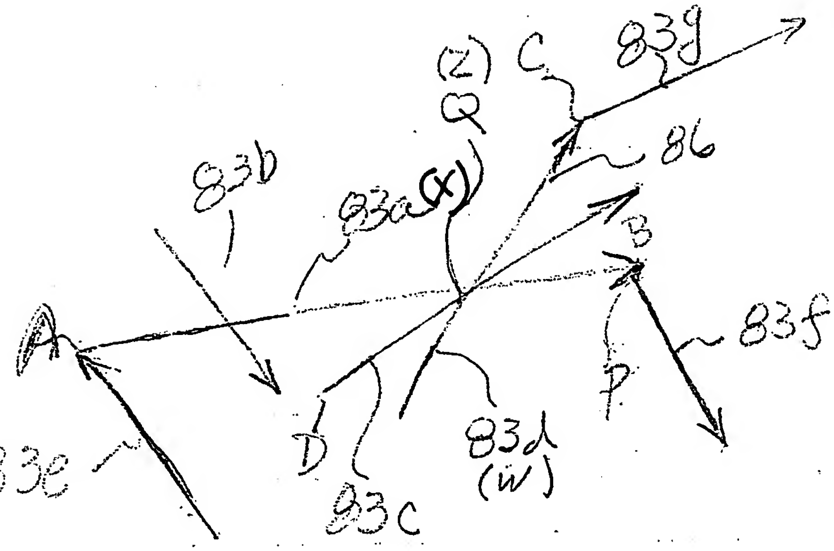
(Z)
Q
C
83g
86
83b
83a
(X)
B
A
83f
P
83e
D
83c
83d
(W)

FIG 8A

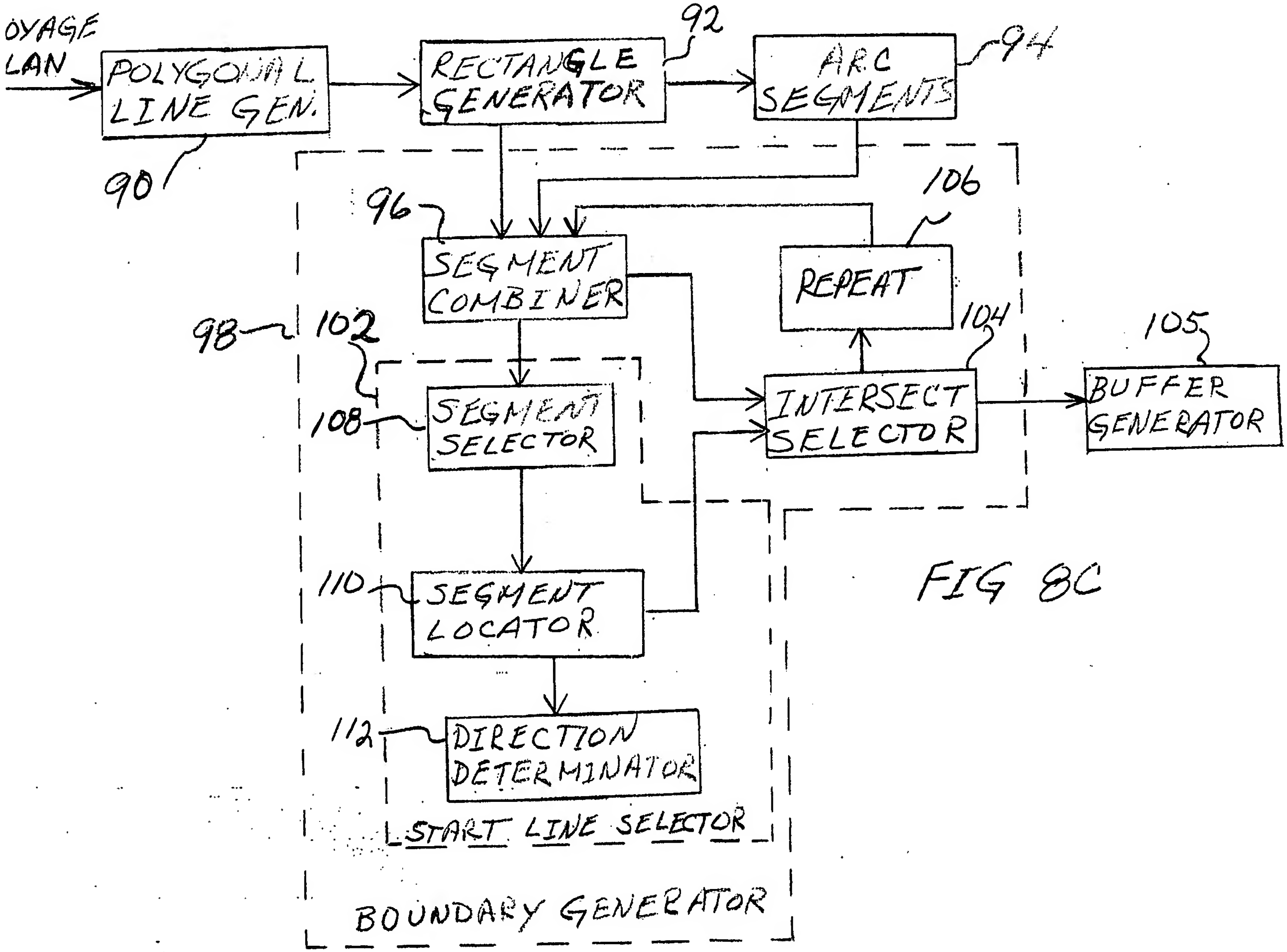
OYAGE
LAN
POLYGONAL
LINE GEN.
90
RECTANGLE
GENERATOR
92
ARC
SEGMENTS
94
96
SEGMENT
COMBINER
106
REPEAT
98
102
104
105
INTERSECT
SELECTOR
BUFFER
GENERATOR
108
SEGMENT
SELECTOR
110
SEGMENT
LOCATOR
112
DIRECTION
DETERMINATOR
START LINE SELECTOR
BOUNDARY GENERATOR

FIG 8C

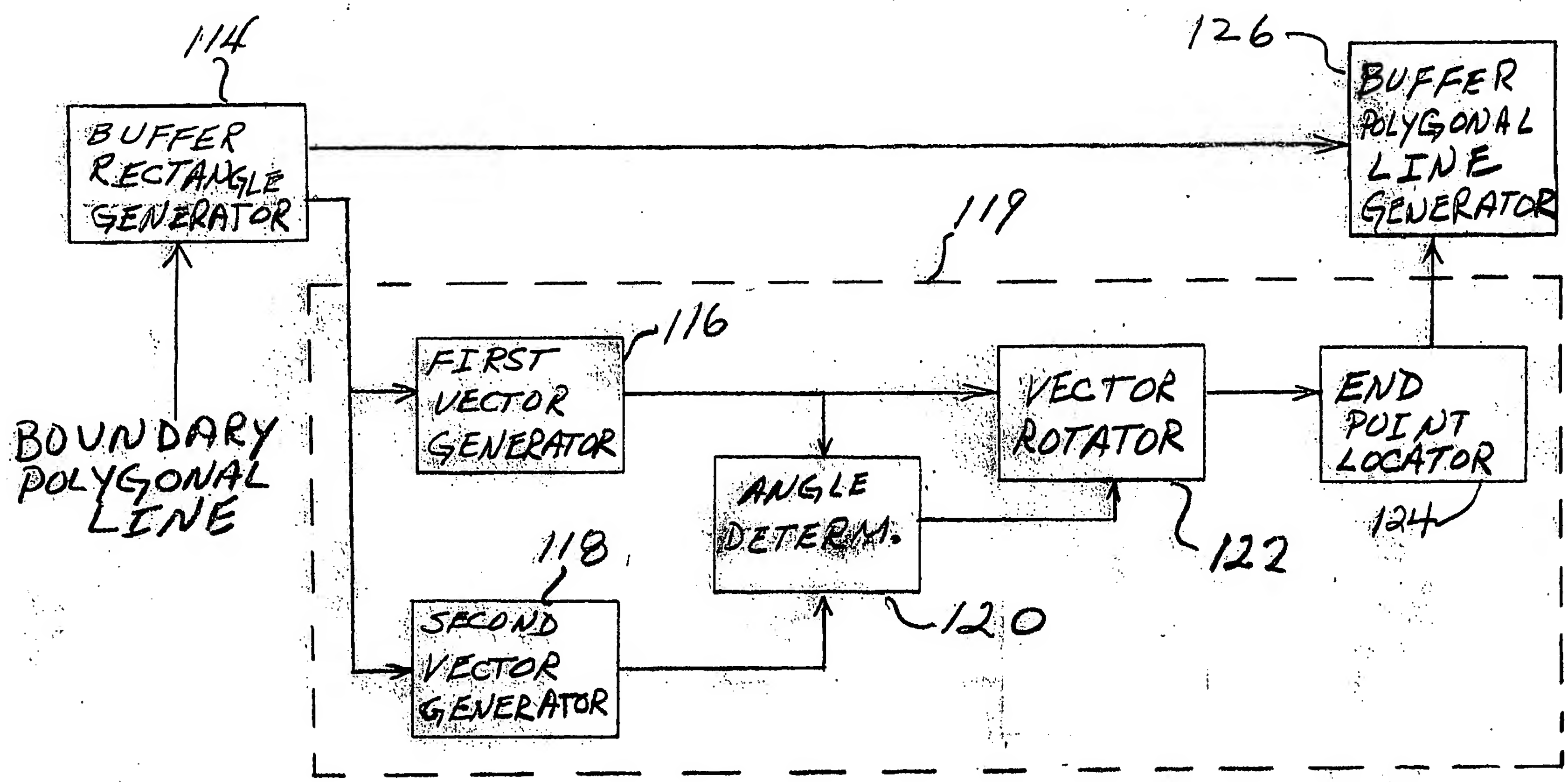
114
BUFFER RECTANGLE GENERATOR
126
BUFFER POLYGONAL LINE GENERATOR
119
116
FIRST VECTOR GENERATOR
VECTOR ROTATOR
END POINT LOCATOR
BOUNDARY POLYGONAL LINE
ANGLE DETERM.
124
118
122
SECOND VECTOR GENERATOR
120

FIG 10